

Convolutional neural networks for leaf image-based plant disease classification

Sachin B. Jadhav¹, Vishwanath R. Udupi², Sanjay B. Patil³

¹Department of Electronics and Telecommunication Engineering, Bharati Vidyapeeths College of Engineering Kolhapur, (Research-Scholar -VTU), Belagavi, Karnataka, India

²Maratha Mandala Engineering College, Belagavi, Karnataka, India

³Shri. Chattrapati Shivajiraje College of Engineering, Bhore, Pune, India

Article Info

Article history:

Received Jul 5, 2019

Revised Oct 5, 2019

Accepted Oct 19, 2019

Keywords:

AlexNet CNN

Deep CNN

DensNet201 CNN

Disease classification

GoogleNet CNN

Machine learning

ResNet101 CNN

VGG16 CNN

ABSTRACT

Plant pathologists desire soft computing technology for accurate and reliable diagnosis of plant diseases. In this study, we propose an efficient soybean disease identification method based on a transfer learning approach by using a pre-trained convolutional neural network (CNN's) such as AlexNet, GoogleNet, VGG16, ResNet101, and DensNet201. The proposed convolutional neural networks were trained using 1200 plant village image dataset of diseased and healthy soybean leaves, to identify three soybean diseases out of healthy leaves. Pre-trained CNN used to enable a fast and easy system implementation in practice. We used the five-fold cross-validation strategy to analyze the performance of networks. In this study, we used a pre-trained convolutional neural network as feature extractors and classifiers. The experimental results based on the proposed approach using pre-trained AlexNet, GoogleNet, VGG16, ResNet101, and DensNet201 networks achieve an accuracy of 95%, 96.4%, 96.4%, 92.1%, 93.6% respectively. The experimental results for the identification of soybean diseases indicated that the proposed networks model achieves the highest accuracy.

Copyright © 2019 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Sachin Balkrushna Jadhav,
Department of Electronics and Telecommunication Engineering,
Bharati Vidyapeeths College of Engineering,
Near Chitranagari, Kolhapur, M.S, 416013.
Email: sachinbjadhav84@gmail.com

1. INTRODUCTION

Soybean crops are profoundly affected by diseases, which causes severe losses in the agriculture economy [1]. For instance, bacterial blight, frog eye leaf spot (FLS), and brown spots are the most common diseases that cause considerable damage to crops and a decrease in yield. The proposed pre-trained AlexNet convolutional neural network (CNN) model used for the classification of these three common diseases. Thus, accurate identification and diagnosis of soybean diseases are vital for high crop yield. In the naked eye approach, which is usually preferred by plant pathologists for detecting soybean diseases, subjective bias can occur because of the decision based on the experience and knowledge of experts [2]. In recent years, various soybean diseases, like fungal such as brown spot, frog eye, rust, bacterial such as pustule and blight, and viral such as bean pod mottle virus explored for automatic detection. To obtain accurate diagnosis results, several researchers have deliberated automated soybean disease diagnosis based on digital image processing [3], pattern recognition [4], and computer vision [5].

A few systems able to work on images captured in fields with different conditions developed. Images acquired with a mobile phone, a method detects and classifies two soybean diseases, such as brown spot and frog eye [6]. For fifty testing samples, the K-NN classifier trained with a shape-based feature vector

is shown to identify brown spot and frog eye with 70% and 80% accuracy. A neural network-based system for classifying downy mildew, frog eye, and bacterial pustule infections reports an accuracy of 93.3% [7]. Another study presented a severity grading system using K-means clustering to automatically detect diseases (bacterial leaf blight, Septoria brown spot, and bean pod mottle virus) [8]. The efficacy of the system evaluated by comparing the results with the manual technique. In [9] proposed a method for soybean disease detection based on salient regions and k-means clustering. In [10] proposed a method for detecting brown spot and frog eye, two common soybean diseases; they used shape features and K-nearest neighbors classification. In [11] presented a technique for detecting insect-damaged vegetable soybean using hyperspectral imaging. In [12] focused on hyperspectral images to study the damage caused by the herbicide glyphosate on soybean plants. In [13] reported image processing techniques for quantitatively detecting rust severity from soybean multispectral images. While doing an extensive literature survey of these research work, we found some limitation in disease region segmentation and classification methods using image processing and computer vision techniques.

Limitations:

- Disease detection and segmentation are essential, but the diseases of soybean are involved in the real environment, and traditional segmentation methods such as k-means, color-based segmentation techniques cannot quickly and accurately obtain segmentation results [14].
- Machine learning methods, such as artificial neural networks (ANNs), Decision Trees, K-means, k nearest neighbors, and Support Vector Machines (SVMs), have been applied in image classification, which based on hand-engineered features [15]. Led to the performance of all these approaches depending heavily on the underlying predefined features [16]. The overall classification accuracy is therefore dependent, on the type of image processing and feature extraction techniques used.

This apparent lack of significant advancements partially be explained by some problematic challenges posed by the image processing and computer vision techniques [17]:

- Uncontrolled capture conditions may present characteristics that make the image analysis more difficult.
- The presence of complex backgrounds cannot easily separate from the region of interest (usually leaf and stem).
- Boundaries of the symptoms often are not well defined.
- Traditional color-based segmentation methods like k-means, Fuzzy K-means, cannot quickly and accurately obtain segmentation results.
- The traditional approach for image classification tasks based on hand-engineered feature Performance of all these approaches depending heavily on the underlying predefined features.
- Overtraining of classifiers, on the defined image database, can cause overfitting problem which may leads inaccuracy in result.

However, a recent trend in machine learning, such as deep convolutional neural networks (CNN's), has demonstrated that learned representations are more effective and efficient. The main advantage of representation learning is that algorithms automatically analyze extensive collections of images and identify features that can categorize images with minimum error [18]. Few researchers proposed the use of CNN for leaf recognition and plant disease classification. In [19] presented convolutional neural network models to perform plant and disease, detection, and classification task using simple leaves images of healthy and diseased plants; it achieves an accuracy of 99.53%. In [20] presented AlexNet and VGG16 CNNs models to identify tomato disease, which achieves 97.29% for VGG16 net and 97.49% for AlexNet. In [21] used pre-trained AlexNet CNN, for disease classification using transfer learning approach. The proposed system was able to classify 26 different diseases in 14 crop species using a database of 54,306 images with a classification accuracy of 99.35%. In [22] proposed a system based on CNNs to recognize 10 common diseases which distinguish between rice blast, rice false smut, rice brown spot, rice bakanae disease, rice sheath blight, rice sheath rot, rice bacterial leaf blight, rice bacterial sheath rot, rice seedling blight, and rice bacterial wilt.; it achieves an accuracy of 95.45%.

All these proposed techniques use convolutional neural networks as both feature extractors and classifiers. We noticed that there is scope to enhance network performance by integrating the CNN network with shallow classifier. Hence, we propose two different approaches used to analyze the performance of networks. In the first approach, we use pre-trained convolutional neural networks as feature extractors and classifiers. In the second approach, we used a pre-trained convolutional neural network as feature extractors and trained on shallow machine learning SVM classifier. This study aims to introduce, the supervised machine learning CNN transfer learning as an approach for classifying three soybean plant diseases out of healthy one according to sample leaf images. This study presents main contributions in plant disease classification using pre-trained CNN networks with advanced Adam optimizers descent on an extensive data set for identification of specific disease symptoms in the soybean infected leaves, which could assist plant pathologists in diagnosing diseases. The paper organized as follows. Section 2 describes the image dataset of

soybean leaves with the method and the materials used to experiment. While Section 3 describes the results of the proposed approach with the obtained evolution matrix. Finally, Section 4 concludes the paper by recommending methods for future improvement.

2. RESEARCH METHOD

Classify various categories of soybean plant diseases, an extensive collection of the plant's leaf images is required. The images downloaded from the Plant Village database [23]. In this section, the methodology followed is discussed in detail.

2.1. Materials

2.1.1. Data set

Data on soybean images downloaded from the plant village database. We analyze 1200 infected and healthy images of plant leaves, which have a spread of 4 class labels assigned to them. Figure 1 shows one example each from every crop disease sample from the Plant Village dataset. In the proposed approaches described in this paper, we resize the images to 227×227 pixels for AlexNet and 224×224 pixels for remaining networks. We performed both the model optimization and predictions on these resized images.

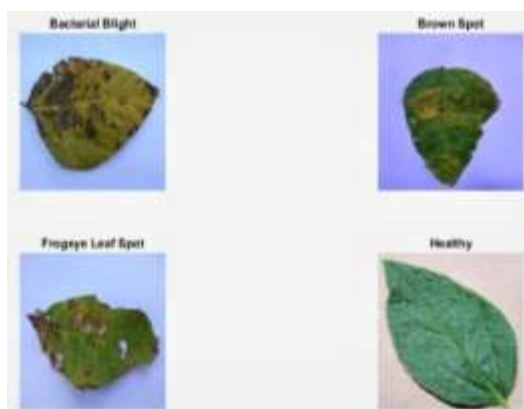


Figure 1. Sample images from the defined database

2.1.2. Performance measure

To know how our proposed approaches perform on new image data, and also to keep a record of if any of our approaches are overfitting, we run all our experiments across an entire range of training and testing data set splits, explicitly in approximate distribution of 80% of the entire dataset used for training, and 20% for testing. The distributed samples per class of the dataset summarized in Table 1.

Table 1. Data set used for training and testing of networks

Sr. No	Disease class	Training Samples	Testing Samples
1	Bacterial Blight	300	70
2	Brown Spot	300	70
3	Frogeye Leaf Spot	300	70
4	Healthy	300	70
	Total	1200	280

2.2. Methods

In this study, we analyze the performance of AlexNet, GoogleNet, VGG16, ResNet101, DensNet201 architectures on the Plant Village dataset by adapting already trained models on the ImageNet dataset using transfer learning. In this technique, we reconfigure the weights of layer fc8 in the case of AlexNet, ResNet101, DensNet201, and of the loss 3 classifier layers in the case of GoogleNet. In the first phase of the study, the preprocessed images applied as input to the CNN network. The proposed networks retrained for classifying the four class categories of leaf objects from the defined disease data set. The last layer was reconfigured and modified to the 4, which is set to the defined number of class categories Figure 3. The four-class categories in this study consisted of three disease classes, namely bacterial blight, brown spot,

and FLS, and one healthy class.

2.2.1. Experiments

We use pre-trained CNN networks for experiments over defined dataset to estimate the effectiveness of our method. Experiment, based on the CNN networks used as both feature extractors and classifiers for the classification. The training epoch adjusted with our proposed method. All of these experiments evaluated under the 5-fold cross-validation strategy. The performance of the proposed network, using the first approach, some training parameters of the CNN networks were modified. The modification included setting the learning rate of the models as 0.0001. The minibatch size set to 64, the number of epochs fixed to 30, and the number of iterations set to 330. The minibatch obtained by splitting the training data set into batches and the gradient descent applied for a network coefficient.

2.2. Architecture of the AlexNet and GoogleNet deep CNN models

The AlexNet, GoogleNet, VGG16, ResNet101, and DensNet201 network tested in the experiment problem, which involved the identification of soybean plant diseases from their leaf images. The convolutional neural network passes a raw image through the network layers and provides a final class as an output. The proposed network consists of 3 convolutional layers, each followed by a max-pooling layer. ReLu activation function applied to the output of every convolutional layer and fully connected layer. The proposed networks consisted of fully connected layers, with each layer network learning to detect different features. Filters then applied to each training image at different resolutions, and the output of each convolved image used as the input to the next layer. Brightness and edge features were detected the complexity of features that uniquely define the leaf object increases as the layers progress. Figure 2 shows the proposed pre-trained AlexNet, and the GoogleNet general CNN model included three main neural layers, namely convolutional layers, pooling layers, and fully connected layers. The three commonly used neural layers discussed as follows [23]:

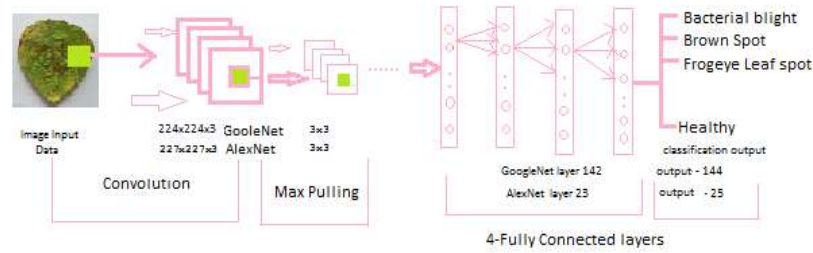


Figure 2. Proposed AlexNet and GoogleNet CNN general architecture

2.2.1. Convolutional layers:

Convolution layers process the input images through a set of convolutional filters, each of which activates certain features from the images. Generally, the convolutional layer output represented by (1)

$$M_j^p = f(\sum_{i \in M_j} M_i^{p-1} * k_{ij}^p + N_j^p) \quad (1)$$

Where p represents the pth layer, k_{ij} denotes convolutional kernel, N_j denotes bias, and M_j denotes a set of input maps. The various parameters of architecture, such as the bias and the weight of the kernel, are typically trained using unsupervised learning approach [14, 19]. The raw input image applied to the convolutional layer through a set of filters, each of which activates certain features from the raw input image. In the convolutional layers, a CNN utilizes various kernels to convolve the whole raw input image as well as the intermediate feature maps, generating various feature maps.

2.2.2. Pooling layers:

Pooling layers simplify the output by performing nonlinear downsampling, which reduces the number of parameters that the network must learn. In stochastic pooling, the probability p should first compute for each region j according to (2)

$$P_i = \frac{\alpha_i}{\sum_{k \in S_j} \alpha_k} \quad (2)$$

Where S_j is pooling region j , F is feature map, and i is every element index inside region j . Stochastic St, is, used in pooling operation for each future map F , the stochastic (St) is expressed by:

$$\alpha_{xy}^{p,k} = St(m, n, x, y) \varepsilon P(\alpha_{m,n}^{p-1,F} w(x, y)) \quad (3)$$

Where $\alpha_{p,k,x,y}$ is the neuron activation at coordinate (x, y) in feature map F in p^{th} layer, $w(x, y)$ is the weighing function.

2.2.3. Fully connected layers:

Fully connected layers “flatten” the network’s 2D spatial features into a 1D vector that represents image level features for classification purposes.

2.3. Image preprocessing and labeling

To improve the recognition accuracy of the proposed models during feature extraction, the final images intended to used as the training and testing data sets for the proposed deep neural network classifier preprocessed for consistency. A total of 1200 soybean leaf sample images, preprocessed to input image dimensions of $227 \times 227 \times 3$ for the AlexNet architecture model and $224 \times 224 \times 3$ for the GoogleNet, VGG16, ResNet101, DensNet201 Network architecture. Then, the preprocessed sample images from the training data set were used to train these CNN architectures. These preprocessed image samples spread of 4 class labels assigned to them. Enhance the recognition accuracy of the proposed models, the conventional ML model training parameters, such as the max epoch, minbatch size, and learning rate, weight optimizer techniques were modified.

2.4. AlexNet and GoogleNet CNN training

Network training involves two stages: a forward stage and a backward stage. First, the main goal of the forward stage is to represent the input image with the current parameters (weights and bias) in each layer. Then, the prediction output is used to compute the loss cost with the ground truth labels. Second, according to the loss cost, the backward stage computes the gradients of each parameter by using chain rules. All the parameters are updated according to the gradients and prepared for the next forward computation. Network learning halted after sufficient iterations of the forward and backward stages. In the feedforward pass stage, we consider a soybean disease multiclass task with N classes and training samples. The squared error function is given by:

$$E^T = \frac{1}{2} \sum_{t=1}^T \sum_{k=1}^N (d_k^t - y_k^t)^2 \quad (4)$$

Where d_k^t is the k^{th} dimension of the t^{th} pattern’s corresponding label, and y_k^t is the value of the k^{th} output layer unit in response to the t^{th} input pattern? We have used supervised learning techniques to train the proposed CNNs to learn the classification of 4 various soybean diseases. Thus, from the image futures, CNNs learned to recognize soybean diseases based on maximized activation neurons with a stochastic response in the next higher layer Regression is applied in multiclass soybean disease classification task. Suppose $H(m)$ and $J(m)$ are defined training dataset, then $\{(H(1), J(1)), \dots, (H(m), J(m))\}$, $J_i \in (1, 2, \dots, k)$. The probability of classifying m as class J is:

$$P(n^{(i)} = J | m^{(i)}; \theta) = \frac{e^{g^T m^i}}{\sum_{p=1}^k e^{g^T m^i}} \quad (5)$$

2.5. Retraining of pre-trained AlexNet and GoogleNet layers

In the pretraining phase, we have used trained deep architectures on a large data set, such as ImageNet, by using powerful machines [24]. The objective of this phase was to initialize network weights for the next phase. We aimed to use the advantages of these pre-trained architectures to enhance the results in the proposed disease classification task. Figure 3 depicts the process of retraining the AlexNet and GoogleNet models from the raw input image with the predicted output probabilities of each disease. The input images of the network were resized to 227×227 pixels for AlexNet and 224×224 pixels for GoogleNet, respectively. The output results represent the probabilities of each disease. We proposed retraining the deep CNN for developing an image classification model from the data set described in Table 1. We retrained these networks to classify four categories of soybean leaf diseases Figure 3 [24]. The following steps were involved in retraining the networks:

1. Loading the pre-trained network
2. Reconfiguring the last three layers to perform a new recognition task
3. Training the model with new data
4. Testing the performance result

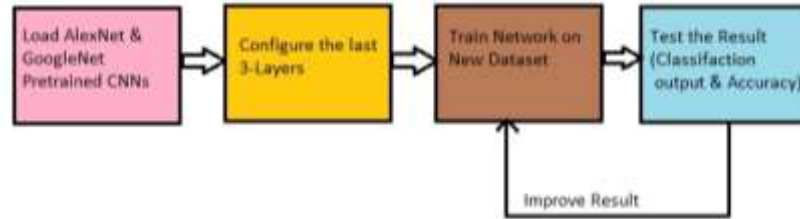


Figure 3. Retraining process of AlexNet & GoogleNet CNNs model

The architectures were reconfigured, modified, and adjusted to support the four defined classes shown in Table 2 and Table 3.

Table 2. Architecture of retrained AlexNet model

Layer	Function	Filter size	Stride
Conv 1	Convolution	11x11x3	4
Pool 1	Max Pooling	3x3	2
Conv 2	Convolution	5x5x48	1
Pool 2	Max Pooling	3x3	2
Conv 3	Convolution	3x3x256	1
Conv 4	Convolution	3x3x192	1
Conv 5	Convolution	3x3x192	1
Pool 5	Max Pooling	3x3	2

Table 3. Architecture of retrained GoogleNet model

Layer	Function	Filter size	Stride
Conv 1	Convolution	11x11x3	4
Pool 1	Max Pooling	3x3	2
Conv 2	Convolution	5x5x48	1
Pool 2	Max Pooling	3x3	2
Conv 3	Convolution	3x3x256	1
Conv 4	Convolution	3x3x192	1
Conv 5	Convolution	3x3x192	1
Pool 5	Max Pooling	3x3	2

3. RESULTS AND DISCUSSION

3.1. Results

3.1.1. Plot of the training progress using Adam optimizer

We aimed to improve the performance accuracy of the model over time. Progress plots obtained for the network training [24]. Figures 4-8 depicts the training progress of the proposed architectures. Our models seem to have improved after the 50th iteration and then increased up to approximately 85-90% validation accuracy for AlexNet, VGG16, ResNet101 and DensNet201 CNN models. It means the network can converge on a solution. We have modified the training options by using advanced optimizers Adam instead of the vanilla Mini-batch gradient descent on a defined image data set and the network configuration as a result of changing the training parameter; we get a much better result more than 92.42% validation accuracy for GoogleNet model. We aimed to improve the performance accuracy of the models over time. Progress plots obtained for the network training.



Figure 4. Plot of training progress for bacterial blight class using AlexNet model



Figure 5. Plot of training progress for bacterial blight class using GoogleNet model



Figure 6. Plot of training progress for bacterial blight class using VGG16 model



Figure 7. Plot of training progress for bacterial blight class using ResNet101model

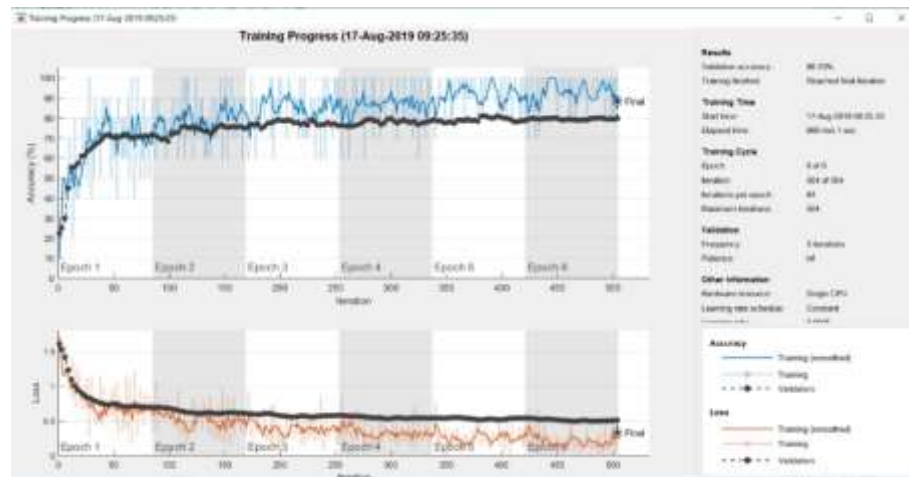


Figure 8. Plot of training progress for bacterial blight class using DensNet201 mode

3.1.2. Confusion matrix and sophisticated confusion matrix

Total 1200 data samples for 4 data classes are considered to train the CNNs model, and a total of 280 data samples considered for testing the performance of the system. From which, there are 14 data sample misclassified, 4 data in class1 misclassified, 3 data in class2 misclassified, and 7 data in class3 misclassified shown in Figure 9 of confusion matrix of the predicted and actual class categories obtained using AlexNet CNN. So, classification accuracy for disease class 1 is 94.3%, disease class 2 is 95.7% disease class 3 is 90%, and non-disease (healthy) class4 is 100%, respectively, which gives overall average accuracy of 95%, shown in Figure 10 of sophisticated confusion matrix for a leaf with Bacterial Blight, leaf with Brown spot, leaf with Frogeye spot and leaf with healthy summarized in Table 4.

Table 4. Classification Result of proposed CNN architectures

Disease Class CNN Architecture	Bacterial Blight		Brown Spot		Frogeye Leaf Spot		Healthy		Accuracy	
	Actual	Predicted	Actual	Predicted	Actual	Predicted	Actual	Predicted	Test Accuracy in (%)	Validation Accuracy in (%)
AlexNet	70	66	70	63	70	63	70	70	95.0	86.6
GoogleNet	70	67	70	65	70	68	70	70	96.4	90.4
VGG16	70	65	70	67	70	68	70	70	96.4	89.7
ResNet101	70	66	70	55	70	67	70	70	92.1	84.4
DensNet201	70	54	70	69	70	69	70	70	93.6	88.3

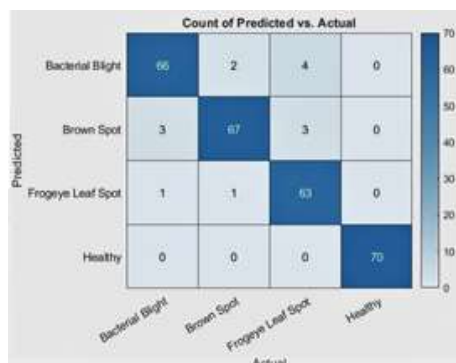


Figure 9. Confusion matrix of AlexNet CNN (Predicted vs. Actual class)

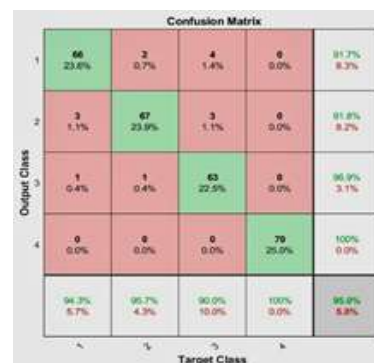


Figure 10. Sophisticated confusion matrix of output class vs. Target class of AlexNet CNN

Similarly, the training data set of the GoogleNet CNN model included 1200 samples for the four data classes and 280 data samples considered for testing the performance of the system. Of these 280 samples, ten misclassified. Three data samples in class 1 misclassified, five data in class 2 misclassified, two data in class 3 misclassified as depicted in the confusion matrix of the GoogleNet CNN Figure 11. Thus, the classification accuracy for class 1 is 95.7%, class 2 is 92.9%, class 3 is 97.1%, and class 4 is 100%, which gives an overall average accuracy of 96.4%, shown in Figure 12 of the sophisticated confusion matrix. The classification results for bacterial blight, FLS, brown spot, and healthy leaves summarized in Table 4.

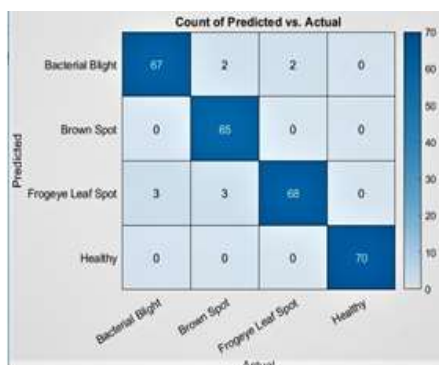


Figure 11. Confusion matrix of GoogleNet CNN (Predicted vs. Actual class)

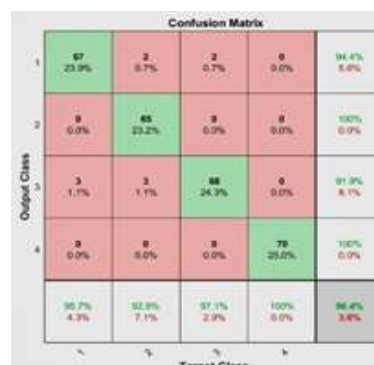


Figure 12. Sophisticated confusion matrix of output class vs. Target class of GoogleNet CNN

Similarly, the training data set of the VGG16 CNN model included 1200 samples for the four data classes and 280 data samples considered for testing the performance of the system. Of these 280 samples, ten misclassified. Five data samples in class 1 misclassified, three data in class 2 misclassified, two data in class 3 misclassified as depicted in the confusion matrix of the VGG16 CNN Figure 13. Thus, the classification accuracy for class 1 is 92.9%, class 2 is 95.7%, class 3 is 97.1%, and class 4 is 100%, which gives an overall average accuracy of 96.4%, shown in Figure 14 of the sophisticated confusion matrix. The classification results for bacterial blight, FLS, brown spot and healthy leaves summarized in Table 4.

Total 1200 data samples for 4 data classes are considered to train the CNNs model, and a total of 280 data samples are supposed to test the performance of the system. Out of 70 test samples, 22 data sample misclassified, 4 data in class1 misclassified, 15 data in class2 misclassified, and 3 data in class3 misclassified shown in Figure Fifteen of the confusion matrix of the predicted and actual class categories obtained using ResNet101 CNN. So, classification accuracy for disease class 1 is 94.3%, disease class 2 is 78.6%, disease class 3 is 95.7%, and non-disease (healthy) class4 is 100%, respectively, which gives overall average accuracy of 92.1%, shown in Figure 16 of sophisticated confusion matrix for the leaf with Bacterial Blight, leaf with Brown spot, leaf with Frogeye spot and leaf with healthy summarized in Table 4.



Figure 13. Confusion matrix of VGG16 CNN (Predicted vs. Actual class)

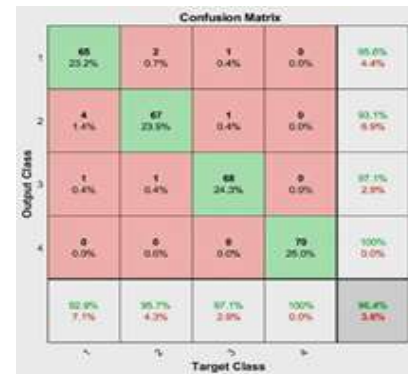


Figure 14. Sophisticated confusion matrix of output class vs. Target class of VGG16 CNN

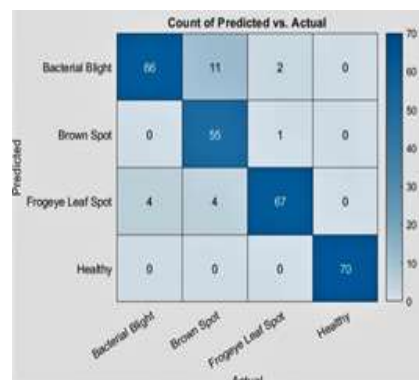


Figure 15. Confusion matrix of ResNet101 CNN (Predicted vs. Actual class)

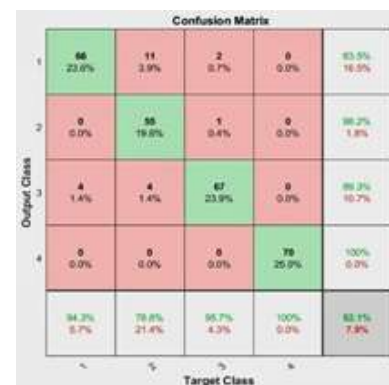


Figure 16. Sophisticated confusion matrix of output class vs. Target class of ResNet101 CNN

Similarly, the training data set of the DensNet201 CNN model included 1200 samples for the four data classes and 280 data samples considered for testing the performance of the system. Of these 280 samples, eighteen misclassified. Sixteen data samples in class 1 misclassified, one data in class 2 misclassified, and one data in class 3 misclassified as depicted in the confusion matrix of the DensNet201 CNN Figure 17. Thus, the classification accuracy for class 1 is 77.1%, type 2 is 98.6%, class 3 is 98.6%, and class 4 is 100%, which gives an overall average accuracy of 93.6%, shown in Figure Eighteen of the sophisticated confusion matrix. The classification results for bacterial blight, FLS, brown spot and healthy leaves summarized in Table 4.

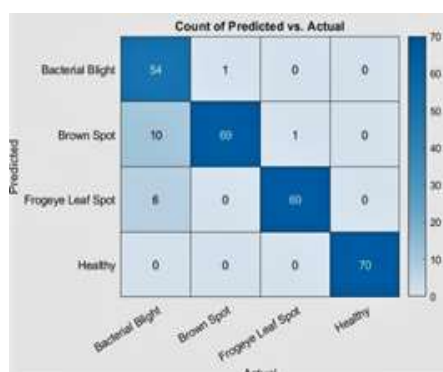


Figure 17. Confusion matrix of DensNet201 CNN (Predicted vs. Actual class)

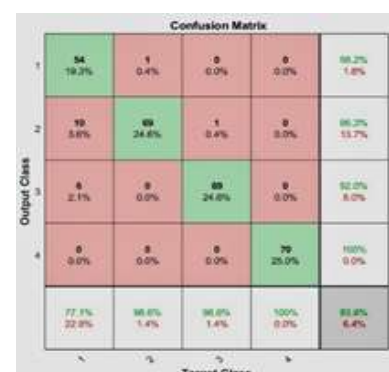


Figure 17. Confusion matrix of DensNet201 CNN (Predicted vs. Actual class)

3.1.3 Training performance and accuracy result

Tables 5-8 presents the training performance of the AlexNet, GoogleNet, VGG16, and ResNet101 CNNs model with hyperparameter details. The tables indicate the elapsed time of training and the overall classification accuracy after testing new data.

Table 5. Training performance of AlexNet CNN

Epoch	Iteration	Time Elapsed (hh: mm: ss)	Mini-batch Accuracy	Validation Accuracy	Mini-batch Loss	Validation Loss	Base Learning Rate
1	1	00:00:53	17.19%	48.96	3.6022	1.6150	0.0010
5	50	00:08:05	92.19%	83.96	0.2225	0.4242	0.0010
10	100	00:14:04	96.88%	91.46	0.1084	0.3669	0.0010
14	150	00:19:33	98.44%	89.38	0.0415	0.4168	0.0010
19	200	00:24:53	100%	89.79	0.0054	0.4898	0.0010
23	250	00:30:15	100%	86.67	0.0250	0.4776	0.0010

Table 6. Training performance of GoogleNet CNN

Epoch	Iteration	Time Elapsed (hh: mm: ss)	Mini-batch Accuracy	Validation Accuracy	Mini-batch Loss	Validation Loss	Base Learning Rate
1	1	00:04:26	21.88%	25.00%	3.5164	3.0742	0.0010
5	50	02:18:13	82.81%	77.92%	0.3887	0.5179	0.0010
10	100	04:36:32	92.19%	83.13%	0.1837	0.4328	0.0010
14	150	06:51:39	100.00%	87.92%	0.0482	0.4593	0.0010
19	200	09:17:46	100.00%	89.38%	0.0300	0.4216	0.0010
23	250	11:42:44	98.44%	88.13%	0.0942	0.3863	0.0010
28	300	15:59:13	100.00%	89.17%	0.0080	0.5225	0.0010
30	330	17:46:21	100.00%	89.79%	0.0016	0.5175	0.0010

Table 7. Training performance of VGG16 CNN

Epoch	Iteration	Time Elapsed (hh: mm: ss)	Mini-batch Accuracy	Validation Accuracy	Mini-batch Loss	Validation Loss	Base Learning Rate
1	1	00:10:07	34.38%	34.79%	2.3549	2.5506	0.0010
5	50	04:30:58	84.38%	78.96%	0.3923	0.4829	0.0010
10	100	09:35:18	92.19%	87.92%	0.1778	0.3865	0.0010
14	150	15:07:29	84.38%	77.29%	0.3655	0.6032	0.0010
19	200	19:24:51	93.75%	89.58%	0.1144	0.3730	0.0010
23	250	29:54:25	100.00%	89.58%	0.0091	0.4220	0.0010
28	300	35:24:51	100.00%	90.00%	0.0019	0.4409	0.0010
30	330	39:15:34	98.44%	90.42%	0.0209	0.4512	0.0010

Table 8. Training performance of ResNet101 CNN

Epoch	Iteration	Time Elapsed (hh: mm: ss)	Mini-batch Accuracy	Validation Accuracy	Mini-batch Loss	Validation Loss	Base Learning Rate
1	1	00:04:26	21.88%	25.00%	3.5164	3.0742	0.0010
5	50	02:18:13	82.81%	77.92%	0.3887	0.5179	0.0010
10	100	04:36:32	92.19%	83.13%	0.1837	0.4328	0.0010
14	150	06:51:39	100.00%	87.92%	0.0482	0.4593	0.0010
19	200	09:17:46	100.00%	89.38%	0.0300	0.4216	0.0010
23	250	11:42:44	98.44%	88.13%	0.0942	0.3863	0.0010
28	300	15:59:13	100.00%	89.17%	0.0080	0.5225	0.0010
30	330	17:46:21	100.00%	89.79%	0.0016	0.5175	0.0010

In this study, 280 samples considered for validation of the AlexNet, GoogleNet, VGG16, ResNet101, and DensNet201 CNNs. A total of 70 samples tested in each disease class category. Figures 19-23 depict the overall classification accuracy of the defined disease class categories when using the proposed CNNs model.

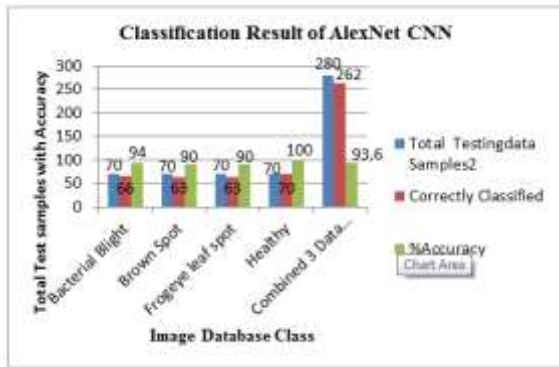


Figure 19. Classification Result of AlexNet CNN

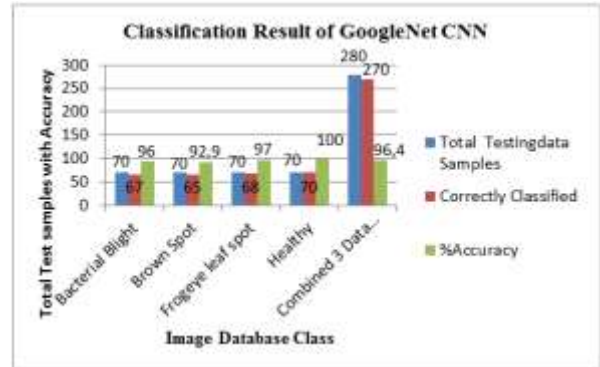


Figure 20. Classification Result of GoogleNet CNN

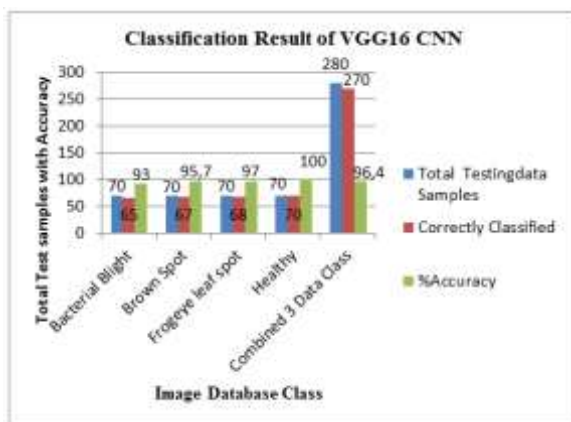


Figure 21. Classification Result of VGG16 CNN

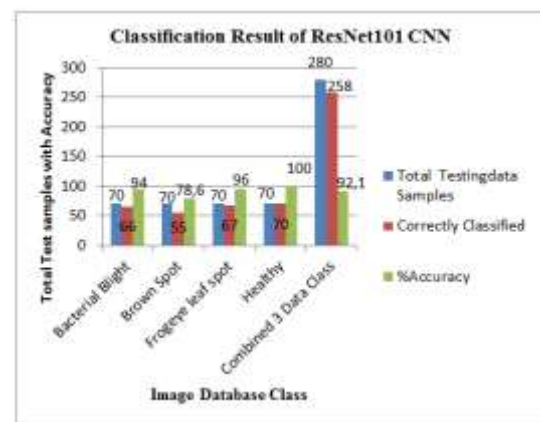


Figure 22. Classification Result of ResNet101 CNN

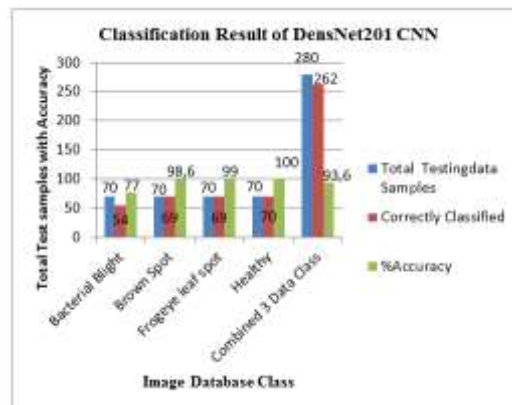


Figure 23. Classification Result of DensNet201 CNN

3.1.4 Comparative analysis with ML system

The performance of the proposed CNN model was compared with that of a previous ML system implemented by [1]. The comparison is presented in Table 9, which indicates that the proposed model outperformed the ML system.

Table 9. Comparative study of Proposed CNNs model with traditional ML system

Particulars	Previous Implemented System by Al-Bashish and et.al.[1]	Proposed AlexNet and GoogleNet CNNs Classifier model
Disease Detected	Early scorch, Cottony mold & Late scorch etc.	Bacterial Blight, Brown spot & Frogeye Leaf spot
No. of samples per class	25	70
Feature extracted	Color & Texture -Manually extracted	Automatically extracted by model
Classifier	Neural Network with Back propagation	VGG16 & GoogleNet CNN Classifier
Accuracy	93%	96.4 & 96.4

4. CONCLUSION

In this study, we proposed a deep learning approach that involved using the AlexNet, GoogleNet, VGG16, ResNet101, and DensNet201 CNNs architectures to build a classifier model for the defined one nondisease and three disease classes (bacterial blight, brown spot, and FLS). The classification accuracies for the proposed pre-trained models were 95%, 96.4%, 96.4%, 92.1%, 93.6% respectively. We used the five fold cross-validation strategy to analyze the performance of networks. We demonstrate the use of the pre trained convolutional neural network as both feature extractors and classifiers outperform than traditional classifiers. The classification performed with the proposed models by modifying various hyperparameters, such as the minibatch size, max epoch, and bias learning rate. Our experimental results indicate that the proposed deep convolutional neural network models achieve the highest accuracy than the machine learning model in soybean disease classification. Future studies can attempt to develop our own CNN model in order to increase the performance rate of the model by varying the minibatch size, bias learning rate, and weight.

ACKNOWLEDGEMENTS

This work supported by Dr.V. S. Patil, Plant Pathology, Zonal Agricultural Research Station, Kolhapur under Mahatma Phule Krishi Vidyapeeth, Rahuri, Maharashtra India.

REFERENCES

- [1] Al Bashish, D M Braik and S Bani Ahmad 2011 Detection and classification of leaf diseases using K-means-based segmentation and neural networks-based classification. *Information Technology Journal*, pp. 267-275.
- [2] Barbedo, Jayme Garcia Arnal 2016 A review on the main challenges in automatic plant disease identification based on visible range images. *Biosystems Engineering*, pp. 52-60.
- [3] Jayme Garcia Arnal Barbedo 2013 Digital image processing techniques for detecting, quantifying and classifying plant diseases, *Springer Plus* 2 (1), pp.660–672.
- [4] Barbedo JGA 2016 A review on the main challenges in automatic plant disease identification based on visible range images, *Biosystem. Eng.*, pp. 52–60.
- [5] Prashar K, Talwar R, Kant C 2015 A review on efficient identification of American cotton leaf diseases through training set. *Int. J. Comput. Appl.*, pp.32–39
- [6] Shrivastava S, Hooda DS 2014 Automatic brown spot and frog eye detection from the image captured in the field. *Am. J. Intell. Syst.*, pp.131–134
- [7] Gharge S, Singh P 2016 Image processing for soybean disease classification and severity estimation. In: Shetty N, Prasad N, Nalini N (eds) *Emerging research in computing, information, communication and applications*, Springer, New Delhi, pp. 493–500.
- [8] Jadhav SB, Patil SB (2015) Grading of soybean leaf disease based on segmented image using k-means clustering. *Int. J. Adv.Res. Electr. Commun. Eng.*, pp.1816–1822
- [9] Gui J, Hao, L Zhang, Q Bao, 2015 A new method for soybean leaf disease detection based on modified salient regions. *Int. J. Multimedia Ubiquitous Engg.*, pp. 45–52.
- [10] Shrivastava, S Hooda, D S, 2014 Automatic brown spot and frog eye detection from the image captured in the field. *Am. J. Intell. Syst.*, pp. 131–134.
- [11] Ma Y, Huang M, Yang, B Zhu Q, 2014. Automatic threshold method and optimal wavelength selection for insect-damaged vegetable soybean detection using hyperspectral images. *Computer Electronics in Agriculture*, pp. 102–110.
- [12] Yao H, Huang Y, Hruska Z, Thomson S J, Reddy K N, 2012. Using vegetation index and modified derivative for early detection of soybean plant injury from glyphosate. *Comput. Electron. Agric.*, pp.145–157.
- [13] Cui D, Zhang, Q Li M, Hartman G L, Zhao Y, 2010 Image processing methods for quantitatively detecting soybean rust from multispectral images. *Biosystem Engineering*, pp.186–193.
- [14] Jayme Garcia Arnal Barbedo 2013 Digital image processing techniques for detecting, quantifying and classifying plant diseases, *Springer Plus*, pp. 1-12.
- [15] Lowe D G, 2004 Distinctive image features from scale-invariant keypoints, *Int. J. Comput. Vis.* pp. 91–110.
- [16] Atabay H. A. 2016 A convolutional neural network with a new architecture applied on leaf classification. *IIOAB J*, pp.226–331.

- [17] Jayme Garcia Arnal Barbedo 2016 A review on the main challenges in automatic plant disease identification based on visible range images, Biosystem Engineering, pp.52-60.
- [18] Reyes A K, Caicedo J C and Camargo J E 2015 Finetuning deep convolutional networks for plant recognition. In CLEF (Working Notes), pp.1-9.
- [19] Konstantinos P Ferentinos 2018 Deep learning models for plant disease detection and diagnosis, Computers and Electronics in Agriculture pp. 311–318.
- [20] Aravind Krishnaswamy Rangarajan, Raja Purushothaman, Anirudh Ramesh 2018 Tomato crop disease classification using pre-trained deep learning algorithm, Procedia Computer Science, pp. 1040–1047.
- [21] Mohanty S P, Hughes D P, and Salathé M 2016. Using deep learning for image-based plant disease detection, Frontiers in plant science, pp.1-10.
- [22] Al Hiary H, S Bani Ahmad, M Reyat, M Braik, and Z A L Rahamneh 2011 Fast and accurate detection and classification of plant diseases, International Journal of Computer Applications, pp.31–38.
- [23] MathWorks 2018 Practical Deep Learning Examples with MATLAB, Mathworks, pp.1-33.
- [24] G E Hinton 2012 A practical guide to training restricted Boltzmann machines, Momentum, pp. 599–619.

BIOGRAPHIES OF AUTHORS



Mr. Sachin. B. Jadhav has pursuing Ph. D from VTU Belagavi and awarded M.E in Electronics and B. E. in Electronics & Telecommunication from Shivaji University Kolhapur. Currently he is working as an Assistant Professor in Department of Electronics & Telecommunication Engineering at Bharati vidyapeeth College of Engineering, Kolhapur. He has keen interest in the field of Image processing.



Dr. Vishwanath R. Udupi has Awarded Ph.D. Graduation and Post-graduation in Electronics Engineering. His field of study is Image processing, Cryptography, Neural Networks, Fuzzy Logic, Bio-Medical Application. His keen interest in application of image processing. He had worked as Professor, Assistant professor and Lecturer under Visveswaraya Technological University, Belagavi as well as Shivaji University Kolhapur. His total education experience is about more than 31 years. Currently working as Principal at Maratha Mandal Engineering College, Belagavi. He has written several technical papers in reputed international Journal and Conferences. He is life member of LM ISTE.



Prof. Dr. Sanjay Bapuso Patil has Awarded Ph. D. in subject of Electronics & Telecommunication, Graduation and Post-graduation in Electronics Engineering. His field of study is Digital Image Processing. His keen interest in application of image processing for —Precision Farming. He had worked as Professor, Assistant professor and Lecturer under Pune as well as Shivaji University Kolhapur. His total education as well as industrial experience is about more than 24 years. Currently working as Principal at Shri. Chattrapati Shivajiraje College of Engineering, Bhore, Pune, India. He has written several technical papers in reputed international Journal and Conferences. He is life member of ISTE.